



09/418772

COFC

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: Albers et al.

Attorney Docket No.: SUN1P223/P3750

Patent: 6,853,868 B1

Issued: February 8, 2005

Title: CROSS-PLATFORM AUDIO FEEDBACK  
FOR GUI COMPONENTS

**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as first-class mail on May 11, 2005 in an envelope addressed to the Commissioner for Patents, P.O. Box 1450 Alexandria, VA 22313-1450.

Signed: \_\_\_\_\_

Kurelia M. Sanchez

**REQUEST FOR CERTIFICATE OF CORRECTION  
OF OFFICE MISTAKE  
(35 U.S.C. §254, 37 CFR §1.322)**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450  
Attn: Certificate of Correction

**Certificate  
MAY 18 2005  
of Correction**

Dear Sir:

Attached is Form PTO-1050 (Certificate of Correction) at least one copy of which is suitable for printing. The errors together with the exact page and line number where the errors are shown correctly in the application file are as follows:

**SPECIFICATION:**

1. Column 3, line 53, change "infiastructure" to --infrastructure--. This appears correctly in the patent application as filed on October 15, 1999, on page 7, line 21.
2. Column 3, line 56, change "taack" to --track--. This appears correctly in the patent application as filed on October 15, 1999, on page 7, line 23.
3. Column 4, line 17, change "GUT object" to --GUI object--. This appears correctly in the patent application as filed on October 15, 1999, on page 8, line 22.
4. Column 4, line 23, change "a GUT" to --a GUI--. This appears correctly in the patent application as filed on October 15, 1999, on page 9, line 3.

MAY 20 2005

5. Column 11, line 37, change "victual machine" to --virtual machine--. This appears correctly in the patent application as filed on October 15, 1999, on page 25, line 12.

**CLAIMS:**

1. In line 2 of claim 4 (column 12, line 28) change "opening system" to --operating system--. This appears correctly in Amendment D as filed on August 11, 2004, on page 2, paragraph 5, line 1.

2. In line 7 of claim 14 (column 14, line 5) change "associated hit" to --associated with--. This appears correctly in Amendment D as filed on August 11, 2004 on page 4, paragraph 2, line 5, as claim 26.

Patentee hereby requests expedited issuance of the Certificate of Correction because the error lies with the Office and because the error is clearly disclosed in the records of the Office. As required for expedited issuance, enclosed is documentation that unequivocally supports the patentee's assertion without needing reference to the patent file wrapper.

It is noted that the above-identified errors were printing errors that apparently occurred during the printing process. Accordingly, it is believed that no fees are due in connection with the filing of this Request for Certificate of Correction. However, if it is determined that any fees are due, the Commissioner is hereby authorized to charge such fees to Deposit Account 500388 (Order No. SUN1P223).

Respectfully submitted,  
BEYER WEAVER & THOMAS, LLP



Steve D Beyer  
Registration No. 31,234

P.O. Box 70250  
Oakland, CA 94612-0250  
650-961-8300

## **DETAILED DESCRIPTION OF THE DRAWINGS**

The present invention will now be described in detail with reference to the preferred embodiments thereof as illustrated in the accompanying drawings. In the following description, numerous specific details are set forth  
5 in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without some or all of these specific details. In other instances, well known process steps and/or structures have not been described in detail in order to not unnecessarily obscure the present invention.

10 A Pluggable L&F (PLAF) apparatus and method is described which emulates the audio functionality of multiple platforms. The platform may be a L&F, operating system (OS) or any other GUI environment. A single audio PLAF with multiple L&F audio functionality is referred to as an Audio L&F (ALF). The Audio L&F includes a set of themes wherein each theme  
15 corresponds to the audio functionality for a particular platform. The theme includes a number of entries wherein each entry provides a mapping between an audio event for a GUI component and the respective audio output.

The Audio L&F apparatus and method may be implemented in Java. In one embodiment, audio functionality may be implemented by utilizing the  
20 conventional PLAF architecture of the Swing components. In another embodiment, the Audio L&F defines and implements additional infrastructure to Java components to permit audio functionality. For example, a Multiplexer may be used to present a single audio L&F as well as track and facilitate

primary L&F changes. In another embodiment of the present invention, the Audio L&F allows the creation of advanced audio user interfaces in Swing to facilitate new applications on the Java platform.

The present invention provides reinforcement of GUI activity through  
5 audio cues. Advantageously, a cross-modality communication of information may be provided to decrease the users' visual load. Further, the present invention permits a flexible alternate presentation of pertinent information as well as utilization of real-world sounds which leverages the users' knowledge of their environment.

10 Additionally, the present invention allows integration of Swing applications into their existing native environments to increase the perceived quality of Java applications. Providing the audio cues that the user expects from their native environment may more fully integrate Java applications into a user's interaction experience. In addition, adding audio feedback to an  
15 interface may increase the perceived quality of the application itself.

FIG. 2 illustrates an audio L&F software configuration 200 inside a running Swing application in accordance with one embodiment of the present invention. The Swing toolkit is included in the Java Foundations Classes (JFC). The software configuration 200 includes a component 202, a UI  
20 Manager 204, a Multiplexer 206, a primary L&F 208 and an Audio L&F 210. These objects communicate to provide audio behavior for the component 202.

The component 202 may be any widget or GUI object whose actions may utilize or lead to an audio event. For example, the component 202 may be a button, scroll bar, frame, etc. The audio event may be any action by the

component 202 which leads to an audio output or function between the computer and the user. By way of example, the event may be the movement of a slider, dragging a GUI object or any other user interface action performed by the component 202. The component 202 may also be capable of multiple  
5 events. By way of example, a mouse button may have over forty audio events within a particular L&F.

A L&F will generally define the user interface for any known component 202 within the L&F. As a result, there will generally be a portion of code responsible for the component 202. The code will determine the look  
10 of the component 202 as well as actions and events performed by the component 202. Attached to this code, a listener 212 may be added for facilitating communication within the audio L&F software configuration 200. The listener 212 may be considered a sensor to an event or action for the component 202.

15 Since the component 202 typically only interacts with a single L&F, the Multiplexer 206 may be used to manage requests and events for multiple and simultaneous L&Fs. To handle the multiple L&Fs, the Multiplexer 206 acts an interface between the component 202 and the current L&F to allow communication therebetween.

20 Conventionally, the primary L&F 208 is the same as the current L&F. With the addition of the Multiplexer 206, the primary L&F 208 may be differentiated from an auxiliary L&F, such as the Audio L&F 210. In this manner, one primary L&F 208 is still maintained, however, there may be a

917 then performs the operation defined by each bytecode as each bytecode is read into interpreter 917. In general, interpreter 917 processes bytecodes 905 and performs operations associated with bytecodes 905 substantially continuously.

5           When a method is called from an operating system 921, if it is determined that the method is to be invoked as an interpreted method, runtime system 919 can obtain the method from interpreter 917. If, on the other hand, it is determined that the method is to be invoked as a compiled method, runtime system 919 activates compiler 915. Compiler 915 then generates  
10 native machine instructions from bytecodes 905, and executes the machine-language instructions. In general, the machine-language instructions are discarded when virtual machine 911 terminates. The operation of virtual machines or, more particularly, Java™ virtual machines, is described in more detail in The Java™ Virtual Machine Specification by Tim Lindholm and  
15 Frank Yellin (ISBN 0-201-63452-X), which is incorporated herein by reference in its entirety.

          Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims.  
20 For example, although the invention has been discussed with respect to the mentioned L&Fs, other L&Fs may be suitably implemented without escaping the nature of the present invention. In addition, the platform independent audio apparatus may be any such service and is not restricted to Java. Alternately, a broad range of potential interaction taxonomies for classifying

This listing of claims will replace all prior versions, and listings of claims in the application:

**Listing of Claims:**

1. (previously presented) A platform-independent audio computer service capable of servicing platform dependent audio events on a first and a second platform, the platform-independent audio computer service comprising:

an audio component capable of receiving as input an audio event, the audio event representing an event which requires an audio response, the audio component running on a first platform serviced by the platform independent audio computer service; and

a software object representing a set of entries, wherein at least one entry of the set of entries is associated with the audio event and a first theme representing an audio cue that maps to the audio event, the first theme including a first set of platform dependent audio fields, each platform dependent audio field associated with at least one platform dependent audio event of the first platform, wherein the first theme is arranged to permit the emulation of the audio event of the first platform in response to the audio event; and

an audio system manager capable of directly or indirectly accessing the software object and audio component, thereby allowing the platform dependent audio event to be emulated using the software object.

2. (previously presented) The audio computer service of claim 1 wherein the audio computer service is implemented in Java programming language.

3. (original) The audio computer service of claim 1 wherein the audio event is a GUI audio event.

4. (original) The audio computer service of claim 1 wherein the first platform is an operating system.

5. (previously presented) The audio computer service of claim 4 wherein the operating system is one of a Windows operating system, a Motif operating system or a Macintosh operating system.

23. (Canceled)

24. (Canceled)

25. (previously presented) A method of servicing platform dependent audio events, by a platform-independent audio computer service, a platform dependent audio field associated with an audio event on a first and a second platform, the method comprising:

providing an audio component capable of receiving as input an audio event, the audio event representing an event which requires an audio response, the audio component running on a first platform serviced by the platform independent audio computer service; and

providing a software object representing a set of entries, wherein at least one entry of the set of entries is associated with the audio event and a first theme representing an audio cue that maps to the audio event, the first theme including a first set of platform dependent audio fields, each platform dependent audio field associated with at least one platform dependent audio event of the first platform, wherein the first theme is arranged to permit the emulation of the audio event of the first platform in response to the audio event; and

providing an audio system manager capable of directly or indirectly accessing the software object and audio component, thereby allowing the platform dependent audio event to be emulated using the software object.

26. (previously presented) A computer-implemented method of accessing, by a platform-independent audio computer service, a platform dependent audio field associated with an audio event from a first platform, the method comprising:

providing a software object representing a set of entries, wherein at least one entry of the set of entries is associated with the audio event and a theme representing an audio cue that maps to the audio event, the theme including a first set of platform dependent audio fields, each platform dependent audio field associated with at least one platform dependent audio event of the first platform, wherein the theme is arranged to permit the emulation of the audio event of the first platform in response to the audio event;

receiving a request for a platform dependent audio function;

importing, using the platform-independent audio computer service, a theme corresponding to the platform dependent audio function including at least one platform



(Also Form PT-1050)

## UNITED STATES PATENT AND TRADEMARK OFFICE CERTIFICATE OF CORRECTION

PATENT NO. : 6,853,868 B1

DATED : February 8, 2005

INVENTOR(S) : Albers et al.

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

**In the Specifications:**

Column 3, line 53, change "infiastructure" to --infrastructure--.

Column 3, line 56, change "taack" to --track--.

Column 4, line 17, change "GUT object" to --GUI object--.

Column 4, line 23, change "a GUT" to --a GUI--.

Column 11, line 37, change "victual machine" to --virtual machine--.

**In the Claims:**

In line 2 of claim 4 (column 12, line 28) change "opening system" to --operating system--.

In line 7 of claim 14 (column 14, line 5) change "associated hit" to --associated with--.

MAILING ADDRESS OF SENDER:

PATENT NO. 6,853,868 B1

Steve D Beyer  
BEYER WEAVER & THOMAS, LLP  
P.O. Box 70250  
Oakland, CA 94612-0250

No. of Additional Copies

1

Burden Hour Statement: This form is estimated to take 1.0 hour to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

MAY 20 2005